LINKgroup

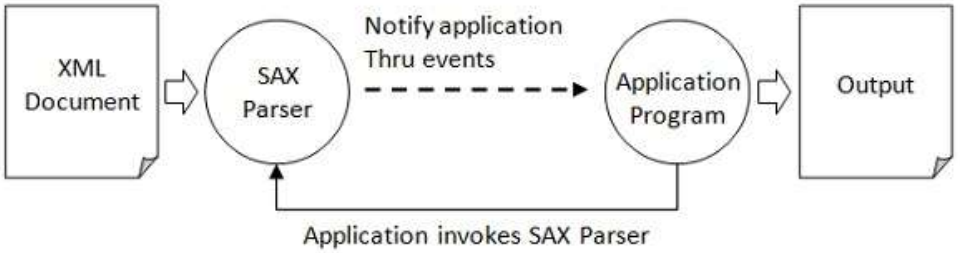
Distance Learning System

Sax parsiranje

Service Application Development

Kako SAX funkcioniše?

- Simple API for XML, SAX, podrazumeva takozvani event based metod rukovanja XML dokumentima. XML dokument se tretira tako što se čita sekvencijalno, tag po tag. Prilikom svakog pristupa i obrade određenog taga dolazi do aktivacije događaja. Na primer, početak čitanja elementa izaziva događaj, kraj čitanja elementa takođe izaziva događaj, čitanje atributa takođe itd.
- Kao rukovaoci događaja definišu se određene callback funkcije u kojima se obrađuje objekat na kome se događaj dogodio (na primer element, ukoliko je u pitanju start ili end element događaj).
- SAX iščitava XML dokument sekvencijalno, pri čemu se, nakon pročitanog taga, nije moguće vratiti nazad na taj tag (Unidirectional Read). Takođe, SAX tehnologija ne podrazumeva mogućnost upisa u XML dokument. Odnosno, ona je Read Only.



Parsiranje dokumenta korišćenjem SAX API-ja

- SAX api podrazumeva tri segmenta:
  - **reader** (xml.sax.xmlreader)
  - **handler** (xml.sax.handler)
  - **utils** (xml.sax.saxutils)

SAX Handler

https://docs.python.org/3/library/xml.sax.handler.html#module-xml.sax.handler

Da bi parsiranje bilo moguće, neophodno je da postoji rukovalac događajima parsiranja. Ovaj događaj je klasa koja nasleđuje klasu ContentHandler

```
class RssHandler(sax.ContentHandler):
    def startElement(self, name, attrs):
        print("Start reading element:",name)
    def endElement(self, name):
        print("End reading element:",name)
    def characters(self, content):
        print("reading content:",content)
    def startDocument(self):
        print("Document read start")
    def endDocument(self):
        print("Document read end")
```

SAX Parsiranje

- Parsiranje se obavlja pomoću metoda **parser (XMLReader)** objekta ili funkcija sax modula
- Metod parse, kao parametar prihvata url koji pokazuje na xml dokument, dok metod **parseString**, kao parametar prihvata str objekat koji sadrži xml dokument

parse funkcija

```
handler = RssHandler()
sax.parse("https://news.yahoo.com/rss/",handler)
```

parser objekat

```
handler = RssHandler()
parser = sax.make_parser()
parser.setContentHandler(handler)
parser.parse("https://news.yahoo.com/rss/")
```

parseString funkcija

```
handler = RssHandler()
xml = request.urlopen("https://news.yahoo.com/rss/").read().decode("utf-8")
sax.parseString(xml,handler)
```

Parsiranje elemenata

- Prepisivanjem metoda za detekciju elemenata, uvodimo sopstvenu logiku parsiranja

```
class RssHandler(sax.ContentHandler):
    def __init__(self):
        self.current_element = ""
    def startElement(self, name, attrs):
        self.current_element = name
    def endElement(self, name):
        self.current_element = ""
```

Parsiranje atributa

- Svaka aktivacija metode startElement, sadrži i parametar attrs
- Attrs sadrži kolekciju atributa
- Attrs implementira metodae mapping protokola
- Attrs sadrži metode za dobavljanje detalja o atributima (getLength, getNames, getType, getValue)

```
def startElement(self, name, attrs):
    for k,v in attrs.items():
        print(k,":",v)
```

Parsiranje sadržaja

- Da bismo pročitali sadržaj samog elementa, potrebno je pregaziti metod characters. Ovaj metod prihvata kao parametar kompletan tekstualni sadržaj nekog elementa

```
def characters(self, content):
    print(content)
```

Selektivno parsiranje

- U prethodnim primerima čitaju se sve vrednosti svih elemenata XML dokumenta. Šta bi bilo da hoćemo da pročitamo samo neke od elemenata XML dokumenta?
- U tom slučaju morali bismo da nekako damo do znanja metodu characters koji je naziv elementa koji trenutno čita

```
def startElement(self, name, attrs):
    self.current_element = name
```

```
def characters(self, content):
    if self.current_element == "title":
        print(content)
```

Vežba 1
(sad-ex01 saxrss.py)

- Na osnovu RSS fida (http://rss.news.yahoo.com/rss/stocks) potrebno je napraviti odgovarajući čitač uz pomoć SAX tehnologije. Ovaj čitač prikazaće sadržaje vesti u sledećem formatu:

  Title: News title
  Link: News link
  Publishing date: News publishing date
  Description: News description